Tutorial 5: Sensor Configuration

The PennArc Team

Introduction

This tutorial will cover the installation of various packages and settings to be configured to get the LiDAR and Inertial Measurement Unit working on Jetson.

Transcript to the Video Tutorial

In this video we will install and configure the Hokuyu LiDAR and Razor IMU. Luckily both have of these have ROS packages readily available and hence the process is quite straight forward. For any other sensor you might want to use in future, you can look up to its package availability in ROS and follow similar steps.

Connect your jetson board to display, keyboard, mouse and an ethernet cable.

On the terminal use the following command to install the Hokuyu node package:

```
sudo apt-get install ros-indigo-hokuyu-node
```

When you connect your lidar to jetson, it generally connects to ACM ports which can be checked using the command :

```
ls /dev/ttyACM*
```

This will list all the ACM ports currently being used. If you are doing this first time, you need to provide specific permissions to the port by using the command :

```
sudo chmod a+rw /dev/ttyACM_
```

This will provide read write permissions for ROS to communicate with the sensor. Now, to check the data coming from LiDAR, as explained in the initial ROS tutorials, run the roscore. On the other terminal , run - rostopic list. This will show all the topics currently being published.

The scan topic is the one we are interested in.

```
rostopic echo /scan
```

These dump of messsages might not make sense now, but these values correspond to the distance sensed by LiDAR at different angles.

We can visualize these laser scans by a GUI interface called Rviz. Unfortunately due to the driver issues Rviz doesn't work on Jetson. If you had setup ROS over Network explained in previous tutorial properly, you can run Rviz over your system that has been setup for the network.

AS you can see these are the laser scans. It might not show up on its own on your system. Insert it using the add button visible here.

Now to install the packages for IMU, go back to Jetson and install the package using command :

```
sudo apt-get install ros-indigo-razor-imu-9dof
```

Lets keep the default parameters untouched, and take a copy for us to make some tweaks

```
roscd razor_imu_9dof/config
cp razor.yaml my_razor.yaml
```

You can calibrate the IMU by following the link provided in the reference. and mention the measured offset values in the newly made file. To confirm the communication of IMU with ROS interface use the launch file:

```
roslaunch razor_imu_9dof razor-pub.launch
```

You can now run ”rostopic list” in another window and look for the /imu topic. ”rostopic echo /imu” will allow you to see the incoming data stream. So the packages for both these sensors are ready. You can follow similar steps to install it on any ROS system. Its better not to clog Jetson’s memory with useless packages and hence you install any extra packages on your laptop.

For example to visualize the IMU data, you need to install the python-visual package. Install the IMU packages on your ROS system as well. Then install the python- visual using :

```
sudo apt-get install python-visual
```

Once done, run the launch file :

```
roslaunch razor_imu_9dof razor-pub-and-display.launch
```

The red board is following the motion of my IMU. Feel free to change the config parameters and play around with this to improve the performance.

References

- Calibrating the Inertial Measurement Unit : Section 7 in http://wiki.ros.org/razor_imu_9dof